

**Obchodní akademie, Vyšší odborná škola a Jazyková škola  
s právem státní jazykové zkoušky Uherské Hradiště**



## **MATURITNÍ PROJEKT**

# **NÁVOD NA VYZKOUŠENÍ CLOUDOVÝCH SLUŽEB**



Obchodní akademie, Vyšší odborná škola  
a Jazyková škola s právem státní jazykové  
zkoušky Uherské Hradiště

Vnitřní předpis OA, VOŠ a JŠ, čj. 025/ORG/2020

Příloha č. 3

### ZADÁNÍ MATURITNÍ PRÁCE INFORMAČNÍ TECHNOLOGIE

Jméno žáka:	Vojtěch Nosek
Téma maturitní práce:	Návod na vyzkoušení cloudových platform
Vedoucí práce:	Ing. Bc. Martin Šimůnek
Oponent práce:	Mgr. Zdenek Hrdina
Způsob zpracování:	<p>Žák prozkoumá dostupné cloudové platformy pro hostování webových aplikací a shrne základní informace o těchto platformách. Předmětem průzkumu by měli být alespoň tři platformy.</p> <p>Žák vytvoří návod, který bude možné použít v rozsahu 4-6 vyučovacích hodin ve 4. ročníku oboru IT. Návod popíše použití jedné zvolené platformy ve čtyřech scénářích:</p> <ol style="list-style-type: none"><li>1. vytvoření statického webu</li><li>2. vytvoření webu nad frameworkem CodeIgniter (rozsahem obdobného souhrnným pracím z předmětu WEB)</li><li>3. vytvoření webu založeného na redakčním systém či obdobném řešení (MediaWiki, WordPress,...)</li><li>4. vytvoření linuxového počítače se vzdáleným přístupem ke konzoli (SSH).</li></ol> <p>Žák připraví návod pro cvičení, který bude rozsahem, přehledností a náročností odpovídat úrovni žáka 4. ročníku oboru IT na naší škole. Návod bude využívat neplacené služby zvolené platformy — bez nutnosti placení alespoň po dobu nutnou pro vyzkoušení návodu.</p> <p>Návod bude umístěn na web (například formou wiki či repozitáře na školním GitHubu). Návod nesmí být pouze přepisem existujících zdrojů.</p> <p>Žák:</p> <ul style="list-style-type: none"><li>• Ověří návod ve výuce a popíše a zapracuje zpětnou vazbu.</li><li>• V práci doporučí vhodné zdroje k dalšímu studiu (například osvědčené tutoriály, videa apod.)</li></ul>



Obchodní akademie, Vyšší odborná škola  
a Jazyková škola s právem státní jazykové  
zkoušky Uherské Hradiště

	<ul style="list-style-type: none"><li>Žák srovná možnosti ostatních cloudových platforem z úvodního průzkumu z pohledu výše uvedených scénářů.</li></ul> <p>Žák připraví plán práce v systému Asana či Trello.</p>
Pokyny k odevzdání:	<p><b>Žák odevzdá práci v tištěné a elektronické podobě</b></p> <ul style="list-style-type: none"><li>Tištěná podoba práce obsahuje uživatelskou a technickou dokumentaci.</li></ul> <p>Tištěnou podobu (v kroužkové vazbě) žák odevzdá v jedné verzi na studijní oddělení školy, místnost 311.</p> <p><b>Elektronická podoba práce obsahuje</b></p> <ul style="list-style-type: none"><li>Dokumentaci ve formátu PDF/A</li><li>Resumé ve formátu PDF/A</li><li>Výsledný projekt, zdrojové soubory a potřebné knihovny pro spuštění projektu</li><li>Prezentaci projektu</li></ul> <p><b>Elektronická podoba</b> práce se nahrává do IS školy dle pokynů vedoucího práce nebo vedení školy.</p> <p>V případě, že se jedná o projekt, na kterém pracovalo více žáků, je povinnou součástí dokumentace podrobné rozdělení činností při práci na projektu.</p>
Kritéria hodnocení:	Hodnocení se skládá z celkové kvality zpracování práce, dokumentace, z kvality prezentace při obhajobě práce, diskuse a z průběžného hodnocení žáka v rámci kontrolních dnů.
Obhajoba projektu	Obhajoba projektu se skládá ze dvou částí - prezentace projektu (včetně podpůrné elektronické prezentace) a diskuse nad řešením. Celková délka obhajoby je 20 minut, délka prezentace projektu by neměla překročit 10 minut.

14. 10. 2021

Datum

Podpis ředitele školy

Obchodní akademie, Vyšší odborná škola  
a Jazyková škola s právem státní  
jazykové zkoušky Uherské Hradiště  
Nádražní 22, 686 01 Uherské Hradiště  
IČO: 60371731, tel.: 572 433 011

### **Prohlášení:**

Souhlasím s tím, že s výsledky mé práce může být naloženo podle uvážení vedoucího maturitní práce a ředitele školy. V případě publikace budu uveden jako spoluautor. Prohlašuji, že jsem na celé maturitní práci pracoval samostatně a veškeré použité zdroje jsem citoval.

V Uherském Hradišti, dne 30.1. 2022

.....

podpis absolventa

## **RESUMÉ**

Cílem tohoto projektu bylo vyzkoušení cloudových služeb Amazon Web Services, Microsoft Azure, Google Cloud Platform a praktické vyzkoušení jejich funkce. Na základě tohoto průzkumu bylo úkolem vypracovat čtyři návody, pomocí kterých si žáci vyzkouší možnosti cloudových služeb.

Zaměřili jsme se na využívání základních služeb, které jsou k dispozici bez placení alespoň po dobu potřebnou k vyzkoušení služby. Za základ jsme považovali službu AWS a dále ji porovnávali se službami cloudů Microsoft Azure a Google Cloud Platform.

# OBSAH

<b>ÚVOD.....</b>	<b>8</b>
<b>1 SEZNÁMENÍ S CLOUDEM .....</b>	<b>9</b>
1.1 Definice a příklad využití .....	9
1.2 Dělení cloudových služeb.....	9
1.2.1 Dělení podle cloudových řešení (Deployment Models) .....	9
1.2.2 Dělení na základě modelů služeb (Service Models).....	10
1.3 Výhody a nevýhody cloudových služeb .....	11
1.4 Modely placení .....	11
1.5 Nejvýznamnější poskytovatelé cloudu .....	12
1.5.1 Amazon Web Services.....	12
1.5.2 Microsoft Azure.....	12
1.5.3 Google Cloud Platform.....	12
<b>2 PRAKTICKÁ ČÁST .....</b>	<b>13</b>
2.1 Tutoriál Nastavení Amazon Web Services .....	13
2.2 Tutoriál „Virtuální stroj na AWS“ .....	15
2.3 Tutoriál „Nahrání statického webu“ .....	17
2.4 Tutoriál „CMS Wordpress“ .....	19
2.5 Tutoriál „Nahrání dynamického webu a zabezpečení“ .....	21
2.6 Postup práce a alternativní řešení .....	23
2.7 Doporučená videa .....	24
<b>ZÁVĚR.....</b>	<b>25</b>
<b>SEZNAM POUŽITÉ LITERATURY .....</b>	<b>26</b>
<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK ..</b>	<b>ERROR! BOOKMARK NOT DEFINED.</b>
<b>SEZNAM OBRÁZKŮ .....</b>	<b>28</b>
<b>SEZNAM TABULEK .....</b>	<b>ERROR! BOOKMARK NOT DEFINED.</b>
<b>SEZNAM PŘÍLOH.....</b>	<b>29</b>
<b>3 DOKUMENTACE K WEBU „ŠKOLA MAPY“ .....</b>	<b>29</b>
OBSAH.....	29
<b>ÚVOD.....</b>	<b>31</b>
<b>4 TECHNICKÁ ČÁST .....</b>	<b>32</b>
4.1 Popis knihoven a frameworku, databáze .....	32

4.1.1	Databáze mapy .....	32
4.1.2	Framework CodeIgniter.....	32
4.1.3	Knihovna Bootstrap.....	32
4.1.4	IonAuth.....	32
4.1.5	DataTables.....	33
4.2	Controller Pages .....	33
4.2.1	Struktura modelu .....	33
4.2.2	Funkce index .....	33
4.2.3	Funkce create.....	34
4.2.4	Funkce school_add .....	34
4.2.5	Funkce pocet_prijatych.....	36
4.2.6	Funkce admitted_add.....	36
4.2.7	Funkce map .....	37
4.2.8	Funkce school_edit .....	37
4.2.9	Funkce school_update .....	38
4.2.10	Funkce admitted_delete.....	38
	Modely.....	39
4.2.11	Model Main_model .....	39
4.2.12	Model Obor .....	40
4.2.13	Model Pocet_prijatych.....	40
4.2.14	Model Skola.....	40
4.3	Routes .....	41
<b>5</b>	<b>UŽIVATELSKÁ DOKUMENTACE .....</b>	<b>43</b>
5.1	Instalace na lokální Apache serveru .....	43
5.2	Popis webu.....	43
5.2.1	Přidávání škol .....	44
5.2.2	Stránka přijatí studenti.....	45
5.2.3	Úprava a mazání údajů o přijatých žácích.....	45
	<b>ZÁVĚR.....</b>	<b>47</b>
	<b>SEZNAM POUŽITÝCH ZDROJŮ .....</b>	<b>48</b>
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>49</b>

## ÚVOD

Maturitní práci jsem si vybral na základě diskuse s mým vedoucím Ing. Šimůnkem. Zajímalo mě, jak vlastně funguje taková cloudová platforma a jaká je obtížnost nasazení webové stránky. Proto jsme se rozhodli zpracovat návody zaměřené na praktické využití se zaměřením na webové stránky a virtuální stroj.

Cloud je slovo, se kterým se v dnešní době setkáváme již běžně. Mezi cloudové služby patří například webový e-mail, cloudová úložiště, a další internetové služby. Ve zkratce cloud znamená globální propojení počítačových sítí, které jsou dostupné z celého světa po internetu.

Hlavním obsahem této maturitní práce je vytvoření a následné zpracování návodu na vybranou cloudovou službu jménem Amazon Web Services. Výsledkem by měl být návod, jenž bude dostupný na webovém hostingu GitHub obsahující čtyři menší návody na využití služeb, které jsou zdarma alespoň po dobu nutnou vyzkoušení. Dále popis obecných vlastností cloudových služeb a seznámení se s nimi. Dalším bodem práce bude průzkum dvou dalších cloudových služeb, konkrétně Microsoft Azure a Google Cloud. Shrnu základní poznatky a možnosti těchto cloudových služeb. Návod bude určený žákům oboru informační technologie, ale i volně dostupný veřejnosti.

Předmětem popisu v návodu je vytvoření tří jednoduchých webových stránek a virtuálního stroje, které poté nasadím na vybranou cloudovou platformu. Prvním návodem bude vytvoření statického webu nasazeného na jedné ze služeb Amazon Web Services. Druhým následuje nasazení webové stránky postavené na frameworku CodeIgniter a třetím zprovoznění webu na systému pro správu obsahu Wordpress a jeho nasazení. Tím posledním bude zprovoznění virtuálního stroje založeném na linuxové distribuci Amazonu, dostupný vzdáleně po internetu pomocí protokolu SSH. V neposlední řadě je vytvoření plánu práce v aplikaci Asana na dobu alespoň čtyř vyučovacích hodin. Jako dodatek doporučím několik videí, které poslouží k samostudiu.



# 1 SEZNÁMENÍ S CLOUDEM

## 1.1 Definice a příklad využití

Na internetu je spousta definic vztahující se ke cloudovým službám, Ve zkrácené podobě znamená, že zprostředkovává a distribuuje výpočetní služby přes internet.

Jedním z příkladů aplikace na cloudové službě je provozovat aplikaci určenou ke komunikaci podobnou messengeru, kterou vlastní firma Meta. K tomu, abychom ji mohli provozovat na světové úrovni bychom potřebovali prostředky, které by nám to umožnili. Jedním z nich jsou cloudové služby, kde si můžeme alokovat výkon podle požadavků na aplikaci. Jedním z typických plusů cloudových služeb jsou náklady, agilita čili přizpůsobení se daným požadavkům aplikace a rychlé nasazení.

**Cloud (Cloud computing)** = v oblasti cloudových služeb se označují IT služby dostupné z internetu. Toto zahrnuje celosvětovou dostupnost prostředků hardwarových tak i softwarových.

## 1.2 Dělení cloudových služeb

### 1.2.1 Dělení podle cloudových řešení (Deployment Models)

**Private (soukromý) cloud** – Jde o cloudové řešení využívané a vlastněné pouze jedinou organizací zahrnující více spotřebitelů. Výpočetní infrastruktura je obvykle vlastněna a spravována soukromou organizací, která ho vlastní, uživateli třetí strany, anebo jejich kombinací. Jedná se o poměrně náročné řešení, jelikož si organizace musí spravovat celé řešení sama a jedná se o technologicky náročné řešení s velkými vstupními náklady, která jsou vhodná pro větší korporace.

**Community (komunitní) cloud** – Jde o cloudové řešení, které je zřízeno a poté sdíleno mezi několika subjekty, nebo několika spotřebiteli z IT organizací. Zařizuje se tehdy, kdy organizace mají společné požadavky (shody, bezpečnostní požadavky). Využívá ho například vláda USA.

**Public (veřejný) cloud** – Jedná se o cloudové řešení, které je dostupné pro širokou veřejnost. Vlastník nemusí mít fyzický server u sebe doma. V případě veřejného cloudu je v rukou poskytovatelů, kteří se o vše postarají. Je méně náročný na administraci. Je to jedno z řešení kdy poskytovatel musí vyhovět široké škále odběratelů. Patří sem řešení od Amazonu, Microsoftu a dalších firem.

**Hybrid (hybridní) cloud** – Tato infrastruktura je spojením dvou řešení, většinou na pomezí privátního a veřejného. Může jít i například o spojení soukromého i komunitního řešení. Tyto dvě infrastruktury fungují samostatně, ale jsou svázány standardizovanou technologií, která umožní přenositelnost dat. Příklad:

1. Vlastník má vybudované datové centrum, kde ukládá citlivá a důležitá data (privátní cloud), ale vše ostatní provozuje na veřejném cloudu.
2. Kombinace privátního a veřejného za pomoci tzv. cloud bursting<sup>1</sup>. Příklad: pokud privátní cloud dosáhne sto procentního využití kapacity svých prostředků, provoz se přesměruje do veřejného cloudu, tím nedochází k přerušení aplikace, popřípadě krátkému výpadku.

**Multicloud** – Multicloud je takový druh cloudového řešení, který využívá více cloudů (např. Microsoft Azure, Google Cloud Platform, AWS), čímž dosahuje většího zabezpečení dat. Umožňuje rozptýřit zátěž, snížit latenci pro uživatele i díky síti datacenter cloudových služeb, kde každá platforma má centra v jiných oblastech a díky tomu se značně zvýší pokrytí.

Multicloud má i své nevýhody. Jednou z nich je například spravování více druhů prostředí, či složitější infrastrukturu, tuto problematiku lze řešit nástroji.

### 1.2.2 Dělení na základě modelů služeb (Service Models)

**Software jako služba (SaaS)** – Ve zkratce znamená pronájem softwaru. Využíváte službu cloudové aplikace, za kterou po určité době zaplatíte poplatek. Příkladem zde může být Microsoft Google Apps a Office 365, který na základě vaší online licence platíte a její obnova vyžaduje připojení k internetu.

**Platforma jako služba (PaaS)** – Tento model je využíván nejčastěji vývojáři. Podporuje vývoj webových aplikací. Vývoj, testování, nasazení, správu a aktualizace. Je to prostředí zprostředkovávající stanice, které poté využíváme. Rozdíl mezi IaaS a PaaS Jedna z nejznámějších platform je Google App Engine, AWS Elastic Beanstalk, Windows Azure, Apache Stratos

**Infrastruktura jako služba (IaaS)** – Tento model poskytuje kompletní IT řešení firmám. Výpočetní technologie (cloud computing), úložiště, síťové a další prostředky, kde spotřebitel nasadí a zprovozní vlastní aplikaci. Příklady: Amazon EC2, Windows Azure, Google Compute Engine

**Funkce jako služba (FaaS)** – FaaS se často pozná a spojuje se slovem „serverless,“ tedy bez serveru. To znamená, že ke spuštění služby nepotřebujeme server a běží nezávisle bez uživatelského prostředí. Umožňuje aplikace spouštět pomocí kontejnerů a vývojář se nemusí starat o další věci. Jedním z příkladů může být AWS Lambda, výpočetní služba bez serveru, která slouží

---

<sup>1</sup> Cloud bursting je konfigurace mezi privátním a veřejným cloudem, která zpracovává požadavky.

ke spouštění kódu dalších aplikací. Další výhody: Platíme jen za to, co využijeme, automatické škálování atd.

### 1.3 Výhody a nevýhody cloudových služeb

Mezi hlavní výhody patří škálovatelnost, díky které si můžeme navyšovat či snižovat kolik RAM, jaké CPU a další parametry bude náš stroj, na kterém aplikace poběží. Toto u serveru, který bychom vlastnili fyzicky není možné. Vše bychom museli zaplatit předem a brát v potaz další kapacitní maxima, kterých provoz dosáhne. Další výhodou je, rychlost spouštění služeb. K založení virtuálního stroje potřebujeme jen pár minut. Platíme jen za to, co využíváme, takže pokud vypneme server, neplatíme nic. K datům můžeme přistupovat z libovolného místa. Ušetříme i na zařízeních, které máme, jelikož pokud se nějaké pokazí, vyměníme za druhé a opět k němu můžeme přistoupit. Co se týče serverů, ty se dají naklonovat čímž zajistíme bez výpadkový provoz služby.

Cloud má ale i své nevýhody, a to, že nemáme přístup k fyzickému serveru a navíc nemůžeme vědět, kdo ho s námi sdílí a kde se přesně nachází. Navíc data nemáme pod kontrolou, pokud tedy nepracujeme s hybridním cloudem.

### 1.4 Modely placení

Amazon Web Services, Microsoft Azure a Google Cloud Platform využívají, stejně jako ostatní cloudy, pay-as-you-go model, známý také jako Pay Per Usage nebo Pay-As-You-Use. Tento model je založen na předplatném a spotřebě. Platí se za délku používání služby, zdrojů (HW – RAM, CPU, GPU atd.), množství uložených dat na disku, a odchozí přenos dat.

Hlavními rozdíly mezi těmito třemi poskytovateli jsou v modelech, které poskytují. Amazon Web Services využívá model hodinový, Microsoft Azure minutový a Google Cloud Platform sekundový.

Model určuje, zda platíme za používání hodinu, minutu či sekundu. Sekundový model je nejlepší z pohledu prostředků, které využíváme v danou dobu. Pokud chceme mít virtuální stroj pronajatý na 5 minut a nechceme zaplatit za celou hodinu, tak zaplatíme pouze za sekundy či minuty, což je výhodnější. K vyzkoušení služeb, které jsou využity v návodech se stejně neplatí, jelikož jsou všechny služby použité zdarma po dobu 12 měsíců, tedy alespoň po dobu nutnou k vyzkoušení.

## 1.5 Nejvýznamnější poskytovatelé cloudu

Mezi jedny z nejvýznamnějších poskytovatelů na trhu, které jsem si vybral jsou Amazon Web Services, Microsoft Azure a Google Cloud Platform. Jako jedni z mnoha nabízejí výběr služeb od nasazení virtuálních strojů, výpočetní operace až po umělou inteligenci.

### 1.5.1 Amazon Web Services

Amazon Web Services je podle definice Amazonu „bezpečná výpočetní platforma, která poskytuje výpočetní výkon, databáze, propojování mezi sítěmi, a disky pro uložení dat.“

AWS je celosvětově využívaná veřejná cloudová služba nejen v zahraničí, ale i u nás, v České republice. Přední zákazníci u nás jsou například Innogy, social bakers, sli.do v zahraničí

AWS nabízí až 24 regionů, kdy každý region je nezávislý a je propojen vysokorychlostní optickými kabely. Nabízí více než 158 Edge locations<sup>2</sup>, aby poskytly co nejnižší latenci, 77 Availability zones<sup>3</sup> ve více než 250 zemích světa, kde v každé zóně má 2 a více datacenter. Je to taky cloud, kterým jsem se nejvíce zabýval.

### 1.5.2 Microsoft Azure

Microsoft Azure je cloudová platforma a umístuje se na druhém místě na trhu. Stejně jako Amazon nabízí přes 200 služeb od nasazení virtuálních strojů až po umělou inteligenci. Tato platforma byla poprvé založena v roce 2010. Nabízí datacentra ve více než 42 zemích světa.

Rozdíl, jaký jsem již zmínil je model placení, který je minutový. Nabízí také různé úrovně zabezpečení virtuálních strojů. Stejně jako AWS, Azure též nabízí 12 měsíců zdarma na vybrané služby.

### 1.5.3 Google Cloud Platform

Google Cloud Platform, jako cloudová platforma začala o něco dříve, v roce 2008, avšak je stále poslední, mezi těmito dvěma soupeři. Své datacentra má ve 29 oblastech, 88 zón a ve více než 200 zemích.

---

<sup>2</sup> Datová centra, která jsou navržena pro minimální možnou latenci. Jsou umístěna blíže k uživateli v hlavních městech.

<sup>3</sup> Místa, kde se nachází seskupení 2 a více datových center.

## 2 PRAKTICKÁ ČÁST

V této části se seznámíme s postupem mé maturitní práce i s úskalími, se kterými jsem se setkal. Dále se zde seznámíme s důvody alternativního řešení.

Zabývám se zde o cloudové služby, jako náhradu za klasické webové hostingy z pohledu nasazení statického a dynamického webu.

### 2.1 Tutoriál Nastavení Amazon Web Services

V tomto tutoriálu se čtenář dozví, jak si nastavit účet Amazon Web Services (AWS). Seznámí čtenáře se základními službami pro správu účtů, nastavení alarmu očekávaných plateb, pokud by chtěl uživatel vyzkoušet služby nad rámec návodu. Dále se seznámí se službami Identity and Access Management Service (IAM), pro správu účtů, nastavení oprávnění pro účet, vytvoření účtu pro běžné užívání, a nastavení další vrstvy zabezpečení Multi Factor Authentication (MFA).

Adresa tutoriálu: [https://github.com/Vojtech550/maturita-cloud/blob/main/AWS\\_nastaveni.md](https://github.com/Vojtech550/maturita-cloud/blob/main/AWS_nastaveni.md)

Ukázky z tutoriálu:

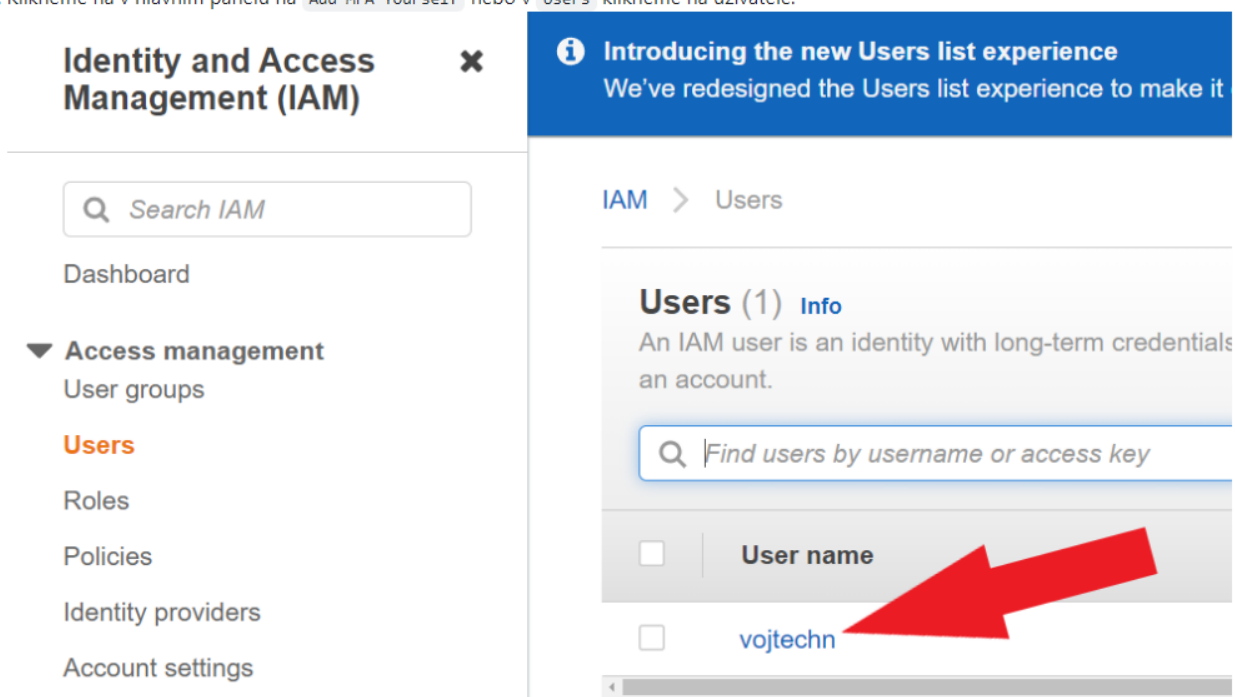
## Obrázek 1 ukázka IAM | zdroj vlastní

### Ověřovací metoda

#### Doplňkové zabezpečení MFA (Multi Factor Authentication)

Doplňkové zabezpečení MFA, neboli Multi Factor Authentication je další vrstvou zabezpečení účtu pro hackerům. Využívá mobilního telefonu s kódem, který se mění.

1. Přejdeme do služby IAM (Identity and Access Management).
2. Klikneme na v hlavním panelu na `Add MFA Yourself` nebo v `Users` klikneme na uživatele.



3. Přejdeme do skupiny `Security Credentials` -> `Assigned MFA device` klikneme na „Manage.“
4. Vybereme metodu „Virtual MFA device“ a dáme „Continue.“
5. Nainstalujeme aplikaci GOogle Authenticator či jinou alternativu: <https://play.google.com/store/apps/details?id=com.google.android.apps.authenticator2&hl=cs&gl=US>

## Obrázek 2 ukázka z IAM 2 | zdroj vlastní

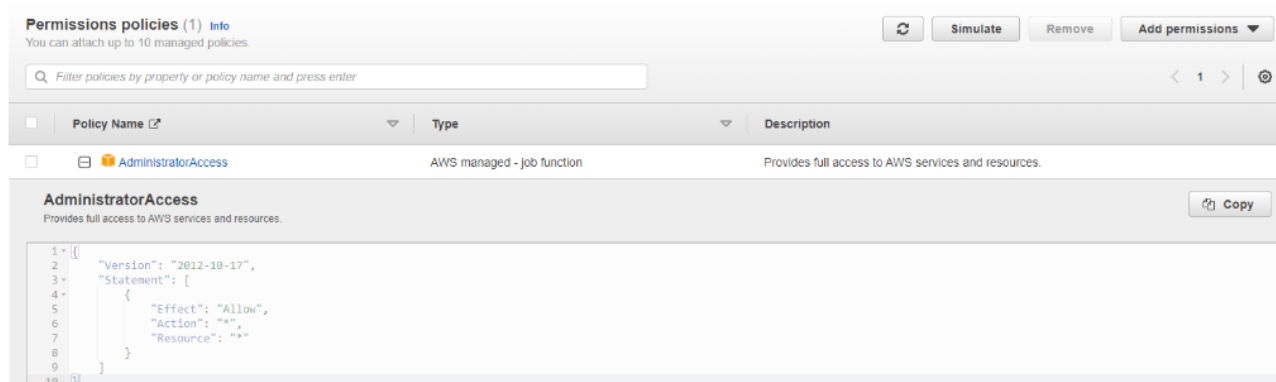
### AWS Identity and Access Management Service (IAM)

#### Vytváření skupiny

Přihlásili jsme se pod root účtem, který má všechna práva k účtu. Kdyby se nám sem ale někdo cizí dostal, mohl by ukrást naše citlivé údaje. K tomu potřebujeme vytvořit uživatele, pod kterým se budeme přihlašovat a zároveň nebude mít přístup k citlivým údajům. A taky si zde usnadníme přihlašování.

Začneme vytvořením skupiny, které přiřadíme práva a následně uděláme uživatele využijeme IAM. Přejdeme d Services, poté Security, v kategorii „Identity & Compliance je služba „IAM“.

Přejdeme do pravého panelu a v „User Groups“ vytvoříme naši skupinu „Create Group“. Pojmenujeme si ji například „Admins“. Dole máme pravidla, tedy oprávnění pro skupiny (Permissions policies). Do vyhledávacího pole napíšeme „AdministratorAccess“, které zaškrtneme. Všechna práva jsou psána v JSONu. V obrázku níže je vidět jednoduchý skript, ve kterém máme vše hlavní zpřístupněno.



## 2.2 Tutoriál „Virtuální stroj na AWS“

Uživatel se v tomto návodu seznámí se službou EC2, díky které si může vytvořit virtuální stroj, který poslouží jako server a v rámci základní úrovně bezplatný. Součástí je i návod na připojení k PuTTY přes protokol SSH. V návodu se uživatel o EC2 dozví jak stroj zprovoznit, nastavení bezpečnostních skupin a nastavení portů a fyzických parametrů virtuálního stroje. Tento tutoriál bude navazovat na ostatní návody, ve kterých využijeme virtuální stroj ke zprovoznění webu.

Tuto službu jsem vybral vzhledem k tomu, že patří k základům na AWS. Podobnou službou je Lightsail, který je též součástí návodů.

Adresa tutoriálu: [https://github.com/Vojtech550/maturita-cloud/blob/main/AWS\\_navod\\_VM.md](https://github.com/Vojtech550/maturita-cloud/blob/main/AWS_navod_VM.md)

Ukázky z tutoriálu:

### Obrázek 3 ukázka vytvoření VM | zdroj vlastní

#### Vytvoření virtuálního stroje

První co musíme udělat je na úvodní obrazovce přejít do „Services“ dále „Compute“ vybrat EC2, kliknout na „Instances“ a „Launch instances“.

Na obrazovce se nám ukáže spoustu distribucí Linuxu a verzí Windows. K tomu, abychom nezvolili nic placeného zaškrtneme „Free tier only“.

**Quick Start**

My AMIs

AWS Marketplace

Community AMIs

☒ Free tier only ⓘ

Poté se zobrazí detaily konfigurace instance, vedle každého pole je kolečko s písmenem „i“, kde je každé pole vysvětlené. Vše necháme tak jak je a klikneme na „Next“. V nabídce s tagy (Add Tag) si lze označit náš virtuální stroj štítkem. Následuje Configure Security Group, kde nastavujeme port a další nastavení ohledně přístupu do Linuxu.

Type (typ portu SSH, TCP..), Protokol (TCP), Port Range (výchozí 22 pro připojení přes protokol SSH), source (definuje IP adresy, které se mohou připojit na danou instanci našeho virtuálního OS), Description (popis), and (Security Group name), nebo si ho přejmenujeme.

1. Choose AMI2. Choose Instance Type3. Configure Instance4. Add Storage5. Add Tags6. Configure Security Group7. Review

**Step 6: Configure Security Group**  
A security group is a set of firewall rules that control the traffic to your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a new security group

☐ Select an existing security group

Security group name:   
Description:

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ	Description ⓘ
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

Add Rule

**Warning**  
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Klikneme na „Next“ a AWS se nás zeptá na vytvoření párů klíčů privátního a veřejného, jelikož máme přístup zabezpečen protokolem SSH.



## Obrázek 4 ukázka připojení přes PuTTY | zdroj vlastní

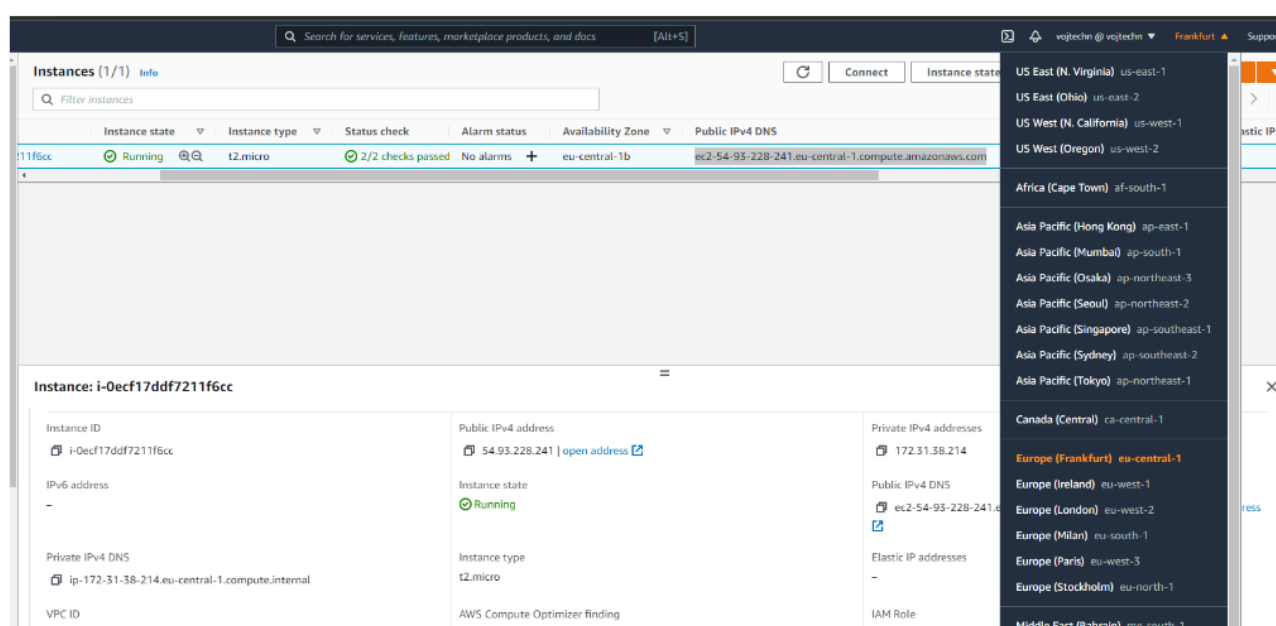
### Připojení k instanci linuxu přes Putty

Odkaz ke stažení programu Putty: <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

1. Po nainstalování spustíme program Puttygen, který nám konvertuje náš stáhnutý SSH klíč k instanci Linuxu. Převeďte ho z formátu „.pem“ do „.ppk“. Soubor s příponou „.pem“ si otevřeme například v aplikaci Notepad ve Windowsu.
2. Zkopírujeme náš klíč do textového souboru a uložíme do formátu „.txt.“
3. Poté klikneme v Putty Key Generatoru na Load, kde klíč v „.txt“ formátu najdeme a po kliknutí na „Save private key“ a uložíme.
4. Následujícím krokem je program Putty. Do kolonky „Host Name (or IP address)“ zadáme naši Host Name/IP adresu naší instance, ke které se chceme připojit. Přejdeme na do EC2 služby a klikneme na „Instances“.

Pozor!!! Musíme se přepnout na region, ve kterém jsme si instanci vytvořili. Nejbližší datacenterum se nechází ve Frankfurtu v Německu, kde jsem si instanci Linuxu vytvořil. Poté nás zajímá **Public IPv4 DNS**. V kolonce v Putty musíme zadat `ec2-@user`, protože se napojujeme na instanci z uživatelského účtu.

Nezapomeňte, že ve výchozím nastavení máme veřejnou IP adresu, která se po vypnutí a smaže a po jejím opětovném spuštění se změní.



## 2.3 Tutoriál „Nahrání statického webu“

Zde se uživatel seznámí se službou Bucket S3, která slouží jako úložiště s krátkodobým přístupem, ale zároveň je jednoduchým řešením, jak provozovat statické stránky. Samozřejmě alternativ je několik, ale služby úložišť se rozdělují dle přístupu k datům, tato služba je nejideálnější.

Dozvíte se o službě Bucket S3 a jak jednoduše nahrávat svůj statický web či využívat jako úložiště. Seznámí vás s limitem této služby i cenou této služby. Vyznačuje se hlavně tím, že je levná. Dále jsem přidal návod s postupem na službě EC2.

Adresa webu:

[https://github.com/Vojtech550/maturitacloud/blob/main/AWS\\_navod\\_static\\_website.md](https://github.com/Vojtech550/maturitacloud/blob/main/AWS_navod_static_website.md)

## Ukázky tutoriálu:

### Obrázek 5 ukázka nastavení S3 Bucket | zdroj vlastní

Default encryption je k zašifrování souborů a Object lock přiřadí zámek k určitým souborům a nepůjdou smazat. Přeskočíme vše ostatní a dáme „Create bucket.“

Rozklikneme si náš bucket a dáme upload. Nahrajeme všechny potřebné soubory pro náš web. V rozhraní máme „add folder“ a „add files“ zvlášť. Po nahrání souborů a složek sjedeme dolů klikneme na „Permission.“ Zde musí být vše veřejně dostupné, takže změníme možnost na „Grant public-read access.“ Properties necháme na standartním, abychom neplatili. Poté klikneme na „Upload.“

▼ Permissions

Grant public access and access to other AWS accounts.

Access control list (ACL)

Grant basic read/write permissions to other AWS accounts. [Learn more](#)

ⓘ AWS recommends using S3 bucket policies or IAM policies for access control. [Learn more](#)

Access control list (ACL)


☒ Choose from predefined ACLs

☐ Specify individual ACL permissions

Predefined ACLs

☐ Private (recommended)  
Only the object owner will have read and write access.

☒ Grant public-read access  
Anyone in the world will be able to access the specified objects. The object owner will have read and write access. [Learn more](#)

 **Granting public-read access is not recommended**  
Anyone in the world will be able to access the specified objects. [Learn more](#)

☒ I understand the risk of granting public-read access to the specified objects.

### Obrázek 6 ukázka nastavení portů EC2 instance | zdroj vlastní

Klikneme na „Next: Add Storage,“ kde si navolíme kapacitu disku. Klikneme na „Next“ dvakrát a nakonfigurujeme „Configure Security Group.“ Zde klikneme na „Select an existing security group.“ Zvolíme si tu, co jsme poprvé zvolili. V mém případě se jmenuje „cloudnewbie.“ Přeskočíme „review details“ a dáme launch. Zvolíme náš pár klíčů, který jsme si vytvořili u předchozí instance Linuxu a zafajfkujeme „I Acknowledge...“ Další problémem bude to, že instanci budeme využívat jako hosting pro náš statický web. Budeme muset změnit Protokol na http.

Assign a security group: ☐ Create a new security group ☒ Select an existing security group

Security Group ID	Name	Description	Actions
sg-0228d902999032623	cloudnewbie	cloudnewbie	<a href="#">Copy to new</a>
sg-03365ea918725a08	default	default VPC security group	<a href="#">Copy to new</a>

Než se nám instance EC2 nastaví a spustí, přejdeme do EC2 služby v „Network & Security“ máme „Security Groups.“ V tomto nastavení nastavíme protokol http (80), abychom mohli provozovat náš web.

Vybereme skupinu, kterou jsme si již dříve vytvořili. V „Inbound rules“ máme po pravé straně „Edit inbound rules.“ Zde přidáme pravidlo (Add rule) a v prvním políčku napíšeme http a dále vybereme Anywhere-IPv4. Přidáme stejné pravidlo pro IPv6 pokud ji budeme používat, ale nemusíme.

HTTP

TCP

80

Anywhere-I...

0.0.0.0/0

Jak můžeme vidět, tak máme celkem 3 protokoly, jeden máme pro IPv6 a druhý pro IPv4.

Inbound rules (3)

Filter security group rules

	Name	Security group rule...	IP version	Type	Protocol	Port range	Source
<input type="checkbox"/>	-	sgr-0c201a35c48effe3f	IPv4	HTTP	TCP	80	0.0.0.0/0
<input type="checkbox"/>	-	sgr-0e77e61a4680fce68	IPv6	HTTP	TCP	80	::/0
<input type="checkbox"/>	-	sgr-07e4f1e3f2100081b	IPv4	SSH	TCP	22	0.0.0.0/0

## 2.4 Tutoriál „CMS Wordpress“

Tento návod seznámí uživatele se službou Lightsail, která se podobá službě EC2. Je to jednodušší varianta virtuálního stroje, na které lze jednoduše zprovoznit systém pro správu obsahu Wordpress. Uživatel se zde dozví, jak si založit web nad CMS<sup>4</sup> Wordpress a jeho základní nastavení. Návod zahrnuje i zabezpečení a zajištění SSL certifikátu zdarma a DNS přes servery třetí strany na adrese <https://www.freenom.com/en/index.html?lang=en>.

Lightsail je odlehčená obdoba EC2 služby, která nenabízí škálování, jako u EC2, která se dá modifikovat. Navíc nefunguje na principu pay-as-you-go modelu. Služba je zdarma po dobu 3 měsíců (750 hodin na měsíc). Více o porovnání EC2 a Lightsail na adrese:

<https://aws.amazon.com/premiumsupport/knowledge-center/lightsail-differences-from-ec2/>

Adresa tutoriálu:

[https://github.com/Vojtech550/maturita-cloud/blob/main/AWS\\_navod\\_wordpress.md](https://github.com/Vojtech550/maturita-cloud/blob/main/AWS_navod_wordpress.md)

<sup>4</sup> Content Management System (Systém pro správu obsahu)

Ukázky z tutoriálu:

### Obrázek 7 ukázka public DNS | zdroj vlastní

#### Doména webové stránky (přes DNS freenom.com) – tato část může být placená!

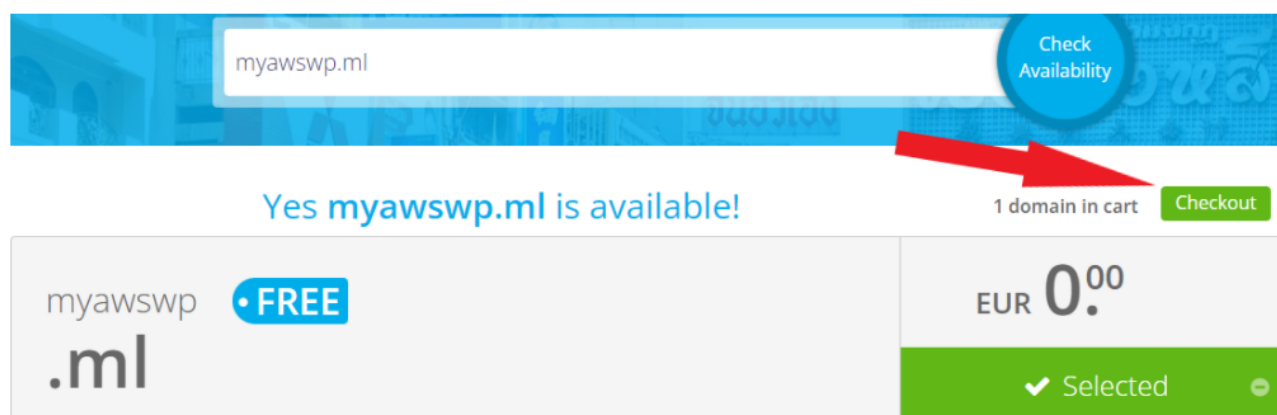
V první řadě chci upozornit, že tato část může být placená. V této části využijeme **public DNS třetí strany**, za kterou bychom za normálních okolností v Amazonu platili. Avšak Amazon si za routy a další účtuje taky. V tomto případě platí, že když smažeme routy do 12 hodin od vytvoření, nebude nás to stát vůbec nic. V opačném případě musíme platit, vyjde to v řádu pár korun.

Dalším krokem bude DNS server. Jelikož Amazon neposkytuje vlastní doménu zdarma, můžeme využít DNS servery zdarma. K tomu využijeme službu Route53 a web Freenom.

Freenom je web, který usnadňuje hledání pro alternativní DNS domény. Jedinou nevýhodou je, že budeme muset využít přípony jako .tk, .ml, .ga a další místo .cz, .com, .net.

DNS je protokol, který překládá adresy z čísel na čtivé a lehce zapamatovatelné názvy. Například Freeweb.cz a další.

Přejdeme na stránku <https://www.freenom.com/en/index.html?lang=en>, kde v navigaci přejdeme do „Services“ a „Register a New Domain.“ Zadáme doménu podle našeho výběru ve tvaru „navez\_webu.přípona“ např.: „myawswp.ml.“ Lze si vybrat ze čtyř přípon: .tk, .ml, .ga, .gq. dostupnost domény a pokud je dostupná, pokračujeme na „Checkout.“



### Obrázek 8 ukázka nastavení DNS | zdroj vlastní

Ted' se podíváme do služby Route53 v konzoli. Na hlavní stránce služby klikneme na „Create hosted zone.“ V „Domain name“ napíšeme adresu, kterou jsme si na freenom vytvořili př. „myawswp.ml.“ a vytvoříme.

<input type="checkbox"/>	Record name ▼	Type ▼	Routin... ▼	Diff... ▼	Value/Route traffic to ▼
<input type="checkbox"/>	myawswp.ml	NS	Simple	-	ns-1646.awsdns-13.co.uk. ns-672.awsdns-20.net. ns-1522.awsdns-62.org. ns-7.awsdns-00.com.
<input type="checkbox"/>	myawswp.ml	SOA	Simple	-	ns-1646.awsdns-13.co.uk. awsdns-hostmaster.amazon.com. 1 7200 900 1209600 86400

Nejdůležitější ze všech částí bude „Value/Route traffic to.“ Označuje servery přes které bude putovat, aby se připojil k naší veřejné doméně. NS znamená Name Server. TTL je time to live, říká, jak dlouho bude DNS záznam uchován v cache lokálního DNS serveru, je nejlepší mít toto číslo co nejnižší, necháme na 300.

Klikneme na Create record -> simple routing. „Record name“ necháme prázdné. Uděláme stejný krok, jen poprvé necháme Record name prázdné a podruhé přidáme „www.“ Do „Value“ vložíme naši statickou IP adresu nebo veřejnou IP adresu, IP adresy oddělujeme klávesou enter.

**Quick create record** [Info](#) [Switch to wizard](#)

▼ Record 1 Delete

Record name [Info](#)  
 myawswp.ml  
Valid characters: a-z, 0-9, ! " # \$ % & ' ( ) \* + , - / : ; < = > ? @ [ \ ] ^ \_ ` { | } . ~

Record type [Info](#)  

A – Routes traffic to an IPv4 address and so...

Value [Info](#)  
 Alias ☐  
Enter multiple values on separate lines.

TTL (seconds) [Info](#)  
  

1m 1h 1d

Recommended values: 60 to 172800 (two days)

Routing policy [Info](#)  

Simple routing

Add another record

## 2.5 Tutoriál „Nahrání dynamického webu a zabezpečení“

Uživatel se dozví jak nahrát vlastní web postavený nad frameworkem a zprovoznit ho na službě EC2. Zároveň je zde popsáno, jak přidat SSL certifikát pomocí Load balanceru a jeho správné nastavení. Load Balancer slouží k „balancování“ dopravy v availability zónách a slouží také jako jedna z možností, jak jej přidělat k webu (možnosti ku příkladu Cloud Front jsou placené). Tutoriál také zahrnuje SSL certifikát a veřejné DNS třetí strany.

Adresa tutoriálu:

[https://github.com/Vojtech550/maturita-cloud/blob/main/AWS\\_navod4\\_CI4\\_web.md](https://github.com/Vojtech550/maturita-cloud/blob/main/AWS_navod4_CI4_web.md)

Ukázky z tutoriálu:

### Obrázek 9 ukázka vystavování certifikátu | zdroj vlastní

1. Najdeme si službu Certificate Manager.
2. Vybereme „Request a certificate“ a zvolíme „Request a public certificate.“
3. V „Domain names“ napíšeme naši webovou stránku, pro mě již vytvořená „myawswp.ml“ „[www.myawswp.ml](http://www.myawswp.ml)“, jelikož pokud chceme udělat web přístupný pod kterýmkoliv formátem, který uživatel zadá, musíme přidat 2 adresy do certifikátu. V Lightsail toto vše bylo řešeno pouze pár příkazy, v EC2 máme nad tím plnou kontrolu.

#### Request public certificate

**Domain names**

Fully qualified domain name [Info](#)

You can add additional names to this certificate. For example, if you're requesting a certificate for "www.example.com", you might want to add the name "example.com" so that customers can reach your site by either name.

**Select validation method** [Info](#)

Select a method for validating domain ownership

☒ **DNS validation - recommended**  
Choose this option if you are authorized to modify the DNS configuration for the domains in your certificate request.

☐ **Email validation**  
Choose this option if you do not have permission or cannot obtain permission to modify the DNS configuration for the domains in your certificate request.

### Obrázek 10 ukázka nastavení Load Balanceru | zdroj vlastní

**Load-Balancer – připojení certifikátu k instanci Linuxu** V rámci služby EC2 je load balancer zdarma na který se musí vázat (nebo k jiné službě), ten potřebujeme k SSL certifikátu.

V konzoli napíšeme „Load Balancer“ a dole v kategorii „Features“ najdeme „Load Balancers“ pro Lightsail. Klikneme na „Create load balancer“ a vybereme „Application load balancer.“ Load Balancer si můžeme libovolně pojmenovat.

#### Nastavení load balanceru

**Název** – zvolíme si libovolný název

**Network mapping** - zde musíme vybrat zóny, kde jsme si vytvořili náš web. Zjistíme v hlavním výběru instancí v EC2 službě v „availability Zone.“ Musíme zvolit minimálně 2 zóny.

Instance state ▼	Actions ▼	Launch instances ▼
Availability Zone ▼	Public IPv4 DNS ▼	Public IPv6
eu-central-1a	ec2-18-196-247-240.eu...	18.196.24

**Listeners and Routing** – zde přidáme kromě protokolu „http“ i „https.“

- V „Listeners and Routing“ klikneme v „http“ protokolu na „Create target group.“ Pokud budeme chtít pouze přesměřovat na https, postačí nám vytvořit skupinu pro https s portem 443 místo portu 80.

## 2.6 Postup práce a alternativní řešení

Průběžný postup práce byl založen na sledování videí ze sociální sítě Youtube a získávání nových znalostí z oblasti cloudových služeb na které jsem se zaměřil. V této oblasti jsem využil základních znalostí o cloudu ze školy, zorientoval se v tématu a zjistil, co hledat. Ač se to zpočátku zdálo obtížné, v září a říjnu jsem začal s podklady z videí, které jsem si sepsal a poté zpracoval úvod ke cloudovým službám čili dělení a další základy, ve kterých by se měl člověk aspoň trochu orientovat.

Zároveň jsem v září dokončil svůj ukázkový web s názvem „Škola mapy“, který byl pokusem na přechod z frameworku CodeIgniter verze 3 na verzi 4, který se ve spoustě věcí odlišuje. Web jsem vypracoval v rámci předmětu webové aplikace. Je dostupný na této webové adrese:

<https://github.com/Vojtech550/maturita-cloud/tree/main/skola-mapy>

K sepsání všech čtyřech návodů jsem využil více videí vzhledem k tomu, že některé videa byly starší či byly jiné postupy než doposud. Též u informací ohledně placených služeb či doplňků služeb, které byly placené jsem čerpal z oficiálního webu Amazonu, ve které jsou uvedeny ceníky. Např. pro službu Route53, která je na krátkou dobu zdarma. Předem přichystaná videa, ze kterých jsem čerpal, jsem uvedl do zdrojů.

Při návodech, ale i se zkoušením jiných cloudových platforem jsem se setkal s problémy. K instalování Apache a MySQL databáze jsem musel přistoupit k nestandardnímu řešení, využil jsem balíček třetí strany, XAMPP. Ten zahrnuje interpret programovacího jazyku PHP, MySQL server a další. Při klasické instalaci Apache a MySQL, jako samostatných balíčků jsem se potkal s problémem v rámci přístupu Apache klienta serveru k webu a instalací repozitáře pro PHP 8.

Všechny návody jsem ze napsal do dokumentu a po přečtení a následně přepsal do značkovacího jazyka markdown pro nahrání na GitHub.

Následovalo vyzkoušení z pohledu scénářů nahrání zprovoznění virtuálního stroje, dynamického webu, statického a webu postaveném na systému pro správu obsahu Wordpress. Zde jsem porovnal cloudy z pohledu cenových modelů, terminologie služeb, která se dá dohledat na internetu, kde Google nabízí veřejný seznam. V tutoriálech AWS jsem služby EC2, Route53, Cloud Watch, S3 Bucket, Certificate Manager, Lightsail, IAM, nastavování účtů a další. (K vyzkoušení v Google Cloud Platform a Microsoft Azure jsem zpracoval menší návody, které mají jen malé odlišnosti od AWS, které jsou v mé práci uvedeny. V únoru a březnu jsem vypracoval dokumentaci a začal na opravě dokumentace a návodů.

Výsledkem práce jsou čtyři návody a porovnání služeb, které je zahrnuté v návodech psané v podobě pro GitHub v markdownu. Vše se dá ověřit přímo v práci pomocí sepsaných návodů, které

vás provedou od základního nastavení až po návody, jak si založit web nad CMS Wordpress, nahrání dynamického či statického webu. Všechny návody jsou k dostání na webové adrese:

<https://github.com/Vojtech550/maturita-cloud>

## 2.7 Doporučená videa

Velice se mi při práci na návodech osvědčilo tohle video, kde jsou vysvětlené principy nejzákladnějších služeb a funkčnost navíc také tutoriál na EC2, který jsem napsal.

<https://www.youtube.com/watch?v=zlmrkM-IvtI>.

Další video může být například toto, které je dost podobné se základy prvního videa.

<https://www.youtube.com/watch?v=Z3SYDTMP3ME>.

Zde je obsaženo přehled základních služeb v 18 minutách, které obsahuje AWS:

<https://www.youtube.com/watch?v=B08iQQhXG1Y>

Pokud se vyznáte v AWS, tak s Google Cloud Platform i Microsoft Azure nebudete mít žádný problém, jelikož služby jsou si dosti podobné, ale s rozdílnou terminologií. Určitě doporučuji videa z kanálu IBM technology, která jednoduše a pochopitelně vysvětlují princip deployment modelů a dalších základů. Např vysvětlení FaaS, PaaS atd.:

<https://www.youtube.com/watch?v=EOIja7yFScs>

Doporučuji videa z kanálu „Be a better dev,“ který má celou škálu videí s AWS a cloudy. Např statický web s Cloudfront (místo něho jsem využil Load Balancer se službou Certificate Manager): <https://www.youtube.com/watch?v=mls8tiiI3uc>



## ZÁVĚR

Ve své práci jsem vypracoval návody na cloudovou službu Amazon Web Services. Službu jsem si vybral na základě návodů na internetu i kvůli tomu, že služby, se kterými jsem chtěl pracovat byly zdarma, tedy aspoň po dobu používání a vyzkoušení. Součástí jsou 4 návody na vybraného cloudového poskytovatele Amazon. Žáci si vyzkouší praktické návody, které můžou sloužit při výukách ve škole. Ať už je to vytvoření virtuálního stroje či přidání webové stránky na cloudovou službu se zabezpečením SSL/TLS, které je v dnešní době jedno z kritérií, jež rozhoduje o tom, zda se web umístí na první či poslední stránce ve výsledcích webového vyhledávače.

Omezení této práce spočívá pouze ve variaci návodů, které jsem uvedl. Jelikož cloudoví poskytovatelé nabízí až 200 služeb, tak jsem vybral nejčastější scénář, který využívají žáci při webových aplikacích, tedy web hosting čili něco, co může posloužit i jako substitute tak praktická dovednost.

Jelikož jsem chtěl zkusit něco nového a praktického, co by mi přineslo základy do budoucna, tak padla volba na toto téma a myslím si, že i když jsem nic podobného v minulosti nezkoušel, tak se až na nějaké problémy vše povedlo. V celku si myslím, že se mi práce podařila, tedy co se týče zpracování návodů, na kterém jsem si dal záležet. I když jsem v jistých případech musel postupovat nestandardně.

Při instalování PHP a Apache serveru jsem narazil na jisté problémy, které jsem musel i přes veškeré mé snahy vynahradit jiným řešením. K instalování jazyka PHP, Apache serveru i MySQL jsem využil balíček XAMPP.

Zdroje na internetu nebyly vždy aktuální, což vedlo k použití více zdrojů a zkombinovat návody dohromady s mým vlastním chápáním. Podle mého názoru je účel návodu, který jsem zpracoval, výstižný a měl by ho pochopit i člověk, který nikdy předtím nedělal s cloudem. Měl by ale znát alespoň základní terminologii v IT.

Zpracování této práce mě obohatilo o nové zkušenosti s cloudovými službami, a i o tom, že tohle je ve skutečnosti daleko obsáhlejší téma, než se probírá ve škole. Také jsem se naučil, že pracovat na jedné práci konstantně je docela obtížné.

## SEZNAM POUŽITÉ LITERATURY

- (1) Algotech: Jak funguje cloud? [online]. [cit. 2022-03-20]. Dostupné z: <https://www.algotech.cz/novinky/2021-10-27-jak-funguje-cloud>
- (2) Youtube: AWS Basics for Beginners Full Course [online]. [cit. 2022-03-20]. Dostupné z: <https://www.youtube.com/watch?v=zlmrkM-IvtI>
- (3) Linuxteaching: Co je Serverless? AWS Lambda a další FaaS [online]. [cit. 2022-03-20]. Dostupné z: [https://cs.linuxteaching.com/article/what\\_is\\_serverless\\_aws\\_lambda\\_and\\_other\\_faas](https://cs.linuxteaching.com/article/what_is_serverless_aws_lambda_and_other_faas)
- (4) Platforma informačních technologií: Ladislav Pešička - Úvod do cloudů [online]. [cit. 2022-03-20]. Dostupné z: <https://pit-plzen.cz/wp-content/uploads/2015/11/CloudyPIT2015.pdf>
- (5) National Institute of Standards and Technology: The NIST Definition of Cloud Computing [online]. [cit. 2022-03-20]. Dostupné z: <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>
- (6) Nimbix: Pay As You Go [online]. [cit. 2022-03-20]. Dostupné z: <https://www.nimbix.net/glossary/pay-go>
- (7) Youtube: AWS Quick Start - Hosting a Static Website on AWS (Demo) [online]. [cit. 2022-03-20]. Dostupné z: <https://www.youtube.com/watch?v=BpFKnPaeloY>
- (8) Youtube: AWS S3 Tutorial for Beginners [online]. [cit. 2022-03-20]. Dostupné z: <https://www.youtube.com/watch?v=L3dYocCSU-E>
- Lightsail:
- (9) Youtube: Host a Wordpress Website on AWS using Amazon Lightsail (For Only \$3.50 a month!) [online]. [cit. 2022-03-20]. Dostupné z: <https://www.youtube.com/watch?v=49aOUHkvlgg>
- (10) Amazon Web Services: Amazon Lightsail or Amazon EC2 Compare Free Cloud Servers [online]. [cit. 2022-03-20]. Dostupné z: <https://aws.amazon.com/free/compute/lightsail-vs-ec2/>
- (11) Amazon Web Services: Amazon Lightsail features [online]. [cit. 2022-03-20]. Dostupné z: <https://aws.amazon.com/lightsail/features/>
- (12) Amazon Web Services: Amazon Lightsail FAQs [online]. [cit. 2022-03-20]. Dostupné z: <https://aws.amazon.com/lightsail/faq/>
- (13) Youtube: AWS Certificate Manager | Get Free SSL Certificate [online]. [cit. 2022-03-20]. Dostupné z: <https://www.youtube.com/watch?v=bWPTq8z1vFY>

- (14) Amazon Web Services: Enabling HTTPS on your WordPress instance in Amazon Lightsail [online]. [cit. 2022-03-20]. Dostupné z: [https://lightsail.aws.amazon.com/ls/docs/en\\_us/articles/amazon-lightsail-enabling-https-on-wordpress](https://lightsail.aws.amazon.com/ls/docs/en_us/articles/amazon-lightsail-enabling-https-on-wordpress)
- (15) Youtube: Amazon EC2 Free Hosting Setup for PHP and MySQL [online]. [cit. 2022-03-20]. Dostupné z: [https://www.youtube.com/watch?v=h04jIAIcBhg&t=178s&ab\\_channel=SrinivasTamada](https://www.youtube.com/watch?v=h04jIAIcBhg&t=178s&ab_channel=SrinivasTamada)

## SEZNAM OBRÁZKŮ

Obrázek 1 ukázka IAM   zdroj vlastní .....	14
Obrázek 2 ukázka z IAM 2   zdroj vlastní.....	15
Obrázek 3 ukázka vytvoření VM   zdroj vlastní .....	16
Obrázek 4 ukázka připojení přes PuTTY   zdroj vlastní .....	17
Obrázek 5 ukázka nastavení S3 Bucket   zdroj vlastní .....	18
Obrázek 6 ukázka nastavení portů EC2 instance   zdroj vlastní .....	19
Obrázek 7 ukázka public DNS   zdroj vlastní .....	20
Obrázek 8 ukázka nastavení DNS   zdroj vlastní .....	21
Obrázek 9 ukázka vystavování certifikátu   zdroj vlastní .....	22
Obrázek 10 ukázka nastavení Load Balanceru   zdroj vlastní.....	22
Obrázek 1 controller a konstruktor .....	33
Obrázek 2 funkce index .....	34
Obrázek 3 funkce create.....	34
Obrázek 4 vyskakovací okénko "Úspěšně přidáno" .....	35
Obrázek 5 funkce pocet_prijatych .....	36
Obrázek 6 funkce admitted_add .....	36
Obrázek 7 funkce map .....	37
Obrázek 8 funkce school_update .....	38
Obrázek 9 funkce delete.....	38
Obrázek 10 model Main_model.....	39
Obrázek 11 metoda select_menu modelu Obor .....	40
Obrázek 12 metody select_skola a select_obor .....	40
Obrázek 13 Ukázka v php.....	41
Obrázek 14 Routes .....	42
Obrázek 15 hlavní stránka.....	44
Obrázek 16 Přidání školy .....	44
Obrázek 17 Úspěšné přidání školy.....	45
Obrázek 18 počet přijatých na škole .....	45
Obrázek 19 editace školy .....	46

# SEZNAM PŘÍLOH

## 2.8 Dokumentace k webu „Škola mapy“

### OBSAH

<b>ÚVOD .....</b>	<b>31</b>
<b>1 TECHNICKÁ ČÁST.....</b>	<b>32</b>
1.1 Popis knihoven a frameworku.....	32
1.1.1 Framework CodeIgniter .....	32
1.1.2 Knihovna Bootstrap .....	32
1.1.3 IonAuth .....	32
1.1.4 DataTables .....	33
1.2 Controller Pages.....	33
1.2.1 Struktura modelu.....	33
1.2.2 Funkce index.....	33
1.2.3 Funkce create .....	34
1.2.4 Funkce school_add.....	34
1.2.5 Funkce pocet_prijatych.....	36
1.2.6 Funkce admitted_add .....	36
1.2.7 Funkce map.....	37
1.2.8 Funkce school_edit .....	37
1.2.9 Funkce school_update.....	38
1.2.10 Funkce admitted_delete .....	38
1.3 Modely .....	39
1.3.1 Model Main_model.....	39
1.3.2 Model Obor.....	40
1.3.3 Model Pocet_prijatych.....	40
1.3.4 Model Skola.....	40
1.4 Routes .....	41
<b>2 UŽIVATELSKÁ DOKUMENTACE.....</b>	<b>43</b>
2.1 Instalace na lokální Apache serveru.....	43
2.2 Popis webu .....	43
2.2.1 Přidávání škol.....	44
2.2.2 Stránka přijatí studenti .....	45

2.2.3 Úprava a mazání údajů o přijatých žácích .....	45
<b>ZÁVĚR.....</b>	<b>47</b>
<b>SEZNAM POUŽITÝCH ZDROJŮ .....</b>	<b>48</b>
<b>SEZNAM OBRÁZKŮ .....</b>	<b>49</b>

## ÚVOD

Cílem bylo vytvořit webovou stránku do předmětu webové aplikace. Stránka bude zobrazovat tabulku se seznamem škol, kdo a kdy byl přijatý. Navíc půjde každá škola zvlášť editovat implementované select menu poslouží pro pohodlný výběr měst a škol. Nebude chybět ani tlačítko pro smazání jednotlivých škol.

Celá stránka je dělaná ve frameworku CodeIgniter 4. Na rozdíl od CodeIgniteru 3 se liší strukturou a jinými metodami. Nová verze má v podstatě podobnou strukturu jako CodeIgniter 3. Problém ovšem nastal u serveru Apache, jelikož jsem musel změnit adresy u linků, nastavit přesměrování na složku public a upravit soubor .htaccess tak, aby vše proběhlo v pořádku a webová stránka se načetla.

## 3 TECHNICKÁ ČÁST

### 3.1 Popis knihoven a frameworku, databáze

#### 3.1.1 Databáze mapy

Autorem této databáze je Michal Slovák.

#### 3.1.2 Framework CodeIgniter

Název: CodeIgniter (verze 4)

Autor: EllisLab

Verze: v4.1.5

Licence: MIT

Odkaz: <https://codeigniter.com/download>

#### 3.1.3 Knihovna Bootstrap

Název: Bootstrap

Autor: Bootstrap team (<https://getbootstrap.com/docs/5.1/about/team/>)

Verze: v4.0.0

Licence: MIT

Odkaz: <https://getbootstrap.com/docs/4.0/getting-started/introduction/>

#### 3.1.4 IonAuth

Název: IonAuth (přihlašovací systém)

Autor: Bened Edmunds

Verze: Pro verzi CI4

Licence: MIT

Odkaz: <https://github.com/benedmunds/CodeIgniter-Ion-Auth>

Port z CodeIgniteru 3.



### 3.1.5 DataTables

Název: DataTables

Autor: SpryMedia Ltd.

Licence: MIT

Odkaz: <https://datatables.net/>

## 3.2 Controller Pages

### 3.2.1 Struktura modelu

Struktura modelu je poměrně jednoduchá, jen od CI3 se liší tím, že každý model, který budeme chtít použít musíme napsat ve stylu `use App\Models\NázevModelu`. Mám jich zde 5: `Main_Model`, `Obor`, `Pocet_prijatych`, `Škola`. V konstruktoru načítám externě přihlašovací systém `IonAuth`.

Obrázek 11 controller a konstruktor

```
<?php

namespace App\Controllers;

use CodeIgniter\Controller;
use App\Models>Main_Model;
use App\Models\Obor;
use App\Models\Pocet_prijatych;
use App\Models\Skola;

class Pages extends BaseController
{
    public function __construct(){
        $this->ionAuth = new \IonAuth\Libraries\IonAuth();
    }
}
```

### 3.2.2 Funkce index

Ve funkci `index` se načítají data a připravují k načtení stránky, konkrétně funkce, která půjde linkovat na hlavní stránku a načte z databáze přes dotaz data z modelu „`Main_Model`.“ Samozřejmě i titulní stránka a další informace se načítají, zkontroluje také, jestli stránka vůbec existuje, i když toho jsem využíval v prvních fázích vývoje. Jako parametr je proměnná `page` s názvem `index`.

Obrázek 12 funkce index

```

public function index($page = 'index')
{
    $model = new Main_Model();
    $data['get_data'] = $model->load_data();
    $data['title'] = 'Databáze škol';
    if ( ! is_file(APPPATH.'/Views/pages/'.$page.'.php')) {
        throw new \CodeIgniter\Exceptions\PageNotFoundException($page);
    }

    echo view('templates/header', $data);
    echo view('pages/'.$page, $data);
    echo view('templates/footer', $data);
}

```

### 3.2.3 Funkce create

Funkce create slouží jako view pro stránku, kde se přidávají školy. Načítá view stránky a také select menu pro města, kterou bere z modelu Škola.

Obrázek 13 funkce create

```

public function create($page = 'school_add')
{
    $data['title'] = 'Přidej školu';
    $skola = new Skola();
    $data['select_mesto'] = $skola->select_mesto();
    echo view('templates/header', $data);
    echo view('pages/'.$page, $data);
    echo view('templates/footer', $data);
}

```

### 3.2.4 Funkce school\_add

Tato funkce se provádí pouze pokud se pokusíme odeslat formulář čili pošle data do databáze, samozřejmě nesmíme zapomenout na vytvoření instance třídy modelu. Na začátku se dotáže, zda jsme přihlášení, za pomoci IonAuth systému. Popřípadě vyhodí chybu, že musíme být administrátorem. Pokud vše proběhne tak jak má, přesměruje nás na index stránku, která se nachází ve složce pages se stránkou: Úspěšně přidáno“.

```

public function school_add()
{
    if (! $this->ionAuth->loggedIn())
    {
        return redirect()->to('/auth/login');
    }
    else if (! $this->ionAuth->isAdmin())
    {
        throw new \Exception('You must be an administrator to view this page.');
```

Obrázek 14 vyskakovací okénko "Úspěšně přidáno"

```

<?php
if (session()->getFlashdata('status'))
{ ?>
    <div class="alert alert-warning alert-dismissible fade show" role="alert">
        <?= session()->getFlashdata('status'); ?>
        <button type="button" class="close" data-bs-dismiss="alert" aria-label="close">
            <span aria-hidden="true">&times;</span>
        </button>
    </div>
<?php
```

### 3.2.5 Funkce pocet\_prijatych

Funkce, která opět zobrazuje a spojuje části navigaci, stránku a patičku stránky. Její součástí je i následné načtení dvou select menu pro výběr školy a oborů.

Obrázek 15 funkce pocet\_prijatych

```
public function pocet_prijatych($page = 'admitted_add')
{
    $data['title'] = 'Přijatí studenti';
    $pocet_prijatych = new pocet_prijatych();
    $data['select_skola'] = $pocet_prijatych->select_skola();
    $data['select_obor'] = $pocet_prijatych->select_obor();
    echo view('templates/header', $data);
    echo view('pages/'.$page, $data);
    echo view('templates/footer', $data);
}
```

### 3.2.6 Funkce admitted\_add

Další POST metoda, kde se na začátku kontroluje, zda je uživatel přihlášený, pokud ano, tak ještě zkontroluje, zda je administrátorem, pokud by chtěl poslat něco do formuláře. Opět posílá správu Úspěšně přidáno, pokud se povedlo do databáze přidat informace o přijatých studentech na školu v daném roce.

Obrázek 16 funkce admitted\_add

```
public function admitted_add(){
    if (! $this->ionAuth->loggedIn())
    {
        return redirect()->to('/auth/login');
    }
    else if (! $this->ionAuth->isAdmin())
    {
        throw new \Exception('You must be an administrator to view this page.');
```

```
    }
    $pocet_prijatych = new Pocet_prijatych();
    $data = [
        'skola' => $this->request->getPost('skola'),
        'obor' => $this->request->getPost('obor'),
        'pocet' => $this->request->getPost('pocet'),
        'rok' => $this->request->getPost('rok'),
    ];
    $pocet_prijatych->save($data);
    return redirect()->to(base_url('pages'))->with('status', 'Úspěšně přidáno');
```

```
}
```

### 3.2.7 Funkce map

Funkce načte model `Main_Model` a od něj pak požádá o data, které potřebuje k zobrazení bodů na mapě a jejich informací ve vyskakovacím okénku. Jako každá funkce, která slouží jen pro načtení frontendu a částečný backend, tak spojuje všechny 3 části webu dohromady (navigaci, samotnou stránku a patičku webové stránky.)

Obrázek 17 funkce map

```
public function map($page = 'maps')
{
    $model = new Main_Model();
    $data['get_data'] = $model->load_data();
    //$data['title'] = ucfirst($page);
    $data['title'] = 'Mapy';
    if ( ! is_file(APPPATH.'/Views/pages/'.$page.'.php')) {
        // Whoops, we don't have a page for that!
        throw new \CodeIgniter\Exceptions\PageNotFoundException($page);
    }

    echo view('templates/header', $data);
    echo view('pages/'.$page, $data);
    echo view('templates/footer', $data);
}
```

### 3.2.8 Funkce school\_edit

Funkce, která slouží pro editování dané školy na hlavní stránce v tabulce. Po kliknutí na tlačítko edit si funkce najde primární klíč (id) školy 5. řádek v kódu, jelikož se upravuje celá tabulka „škola.“ Načítá také view stránky.

```
public function school_edit($id){
    $skola = new Skola();
    $data['title'] = 'Upravit školu';
    $data['select_mesto'] = $skola->select_mesto();
    $data['row'] = $skola->find($id);
    echo view('templates/header', $data);
    echo view('pages/school_edit', $data);
    echo view('templates/footer', $data);
}
```

### 3.2.9 Funkce school\_update

POST funkce, která slouží k aktualizování (ke změně) informací při kliknutí na tlačítko uložit. Využívám místo funkce „save,“ funkci „update,“ vestavenou ve frameworku CodeIgniteru4. Funkce také nejprve musí udělat instanci třídy „Skola“. Pokud ovšem uživatel není přihlášen, nic neudělá.

Obrázek 18 funkce school\_update

```
public function school_update($id){
    if (! $this->ionAuth->loggedIn())
    {
        return redirect()->to('/auth/login');
    }
    else if (! $this->ionAuth->isAdmin())
    {
        throw new \Exception('You must be an administrator to view this page.');
```

### 3.2.10 Funkce admitted\_delete

Funkce podle „id“ smaže záznam přijatých žáků škol za školní rok pokud je uživatel přihlášen a je administrátorem. Načte si nejprve třídu „Skola.“

Obrázek 19 funkce delete

```
public function admitted_delete($id){
    if (! $this->ionAuth->loggedIn())
    {
        return redirect()->to('/auth/login');
    }
    else if (! $this->ionAuth->isAdmin())
    {
        throw new \Exception('You must be an administrator to view this page.');
```

## Modely

### 3.2.11 Model Main\_model

V codeIgniteru jsem zvolil cestu modelů, jelikož v CodeIgniteru 4 nefunguje jako CodeIgniter 3. V proměnné `protected $table` napíšeme název tabulky, poté zvolíme primární klíč, podle kterého se má řídit, tedy mazat, upravovat a další. `returnType`, ve výchozím nastavení vrací array. Snad vůbec nejdůležitější je proměnná `allowedFields`, kde se nacházejí názvy sloupců, se kterými budeme pracovat; používat a načítat je z modelu.

Prázdné proměnné jsem nepotřeboval, ale chtěl jsem vědět, zda je budu vůbec potřebovat, ukázalo se, že ne. Důležitá je proměnná „`useAutoIncrement`.“ I kdybychom měli `autoincrement` v databázi nastavený, tak musí být i v modelu.

Proměnná `softDeletes` je nastavená na `false`, jelikož se vztahuje k codeIgniterské funkci, kterou jsem nevyužil. Místo toho, aby se smazala hodnota ji nastaví na to, co je v proměnné „`delete_at`.“

Na tabulku jsem použil druhý způsob, načítání pomocí query dotazu, kde jsem spojil všechny tabulky (obor, město, `pocet_prijatych`, škola) a poté vypsál všechny hodnoty.

Obrázek 20 model Main\_model

```
namespace App\Models;

use CodeIgniter\Model;

class Main_Model extends Model
{
    protected $table      = 'mesto';
    protected $primaryKey = 'id';

    protected $useAutoIncrement = true;

    protected $returnType     = 'array';
    protected $useSoftDeletes = false;

    protected $allowedFields = ['nazev'];

    protected $useTimestamps = false;
    protected $dateFormat    = 'datetime'; //not needed
    protected $createdField   = 'created_at';
    protected $updatedField   = 'updated_at';
    protected $deletedField   = 'deleted_at';
    // Validation
    protected $validationRules = [];
    protected $validationMessages = [];
    protected $skipValidation = false;
    protected $cleanValidationRules = true;
    // Callbacks
    protected $allowCallbacks = true;
    protected $beforeInsert = [];
    protected $afterInsert = [];
    protected $beforeUpdate = [];
    protected $afterUpdate = [];
    protected $beforeFind = [];
    protected $afterFind = [];
    protected $beforeDelete = [];
    protected $afterDelete = [];

    public function load_data()
    {
        $result = $this->db->query('
        SELECT pocet_prijatych.id AS admitted_id, skola.id AS skola_id, mesto.nazev AS nazev_mesto, obor.nazev AS nazev_obor, pocet_prijatych.pocet AS prijatych,
        pocet_prijatych.rok AS rok_prijeti, skola.nazev AS nazev_skola, skola.geo_lat AS geo_latitude, skola.geo_long AS geo_longitude
        FROM mesto
        INNER JOIN skola ON mesto.id=skola.mesto INNER JOIN pocet_prijatych ON skola.id=pocet_prijatych.skola
        INNER JOIN obor ON pocet_prijatych.obor=obor.id');
        return $result->getResult();
    }
}
```

### 3.2.12 Model Obor

V tomto modelu je všechno stejné až na metodu, kterou využívám k načtení select menu pro výběr oborů pro přidávání přijatých studentů.

Obrázek 21 metoda select\_menu modelu Obor

```
public function select_menu(){
    $select = $this->db->query('SELECT obor.id, obor.nazev FROM obor');
    return $select->getResult();
}
```

### 3.2.13 Model Pocet\_prijatych

V tomto modelu mám dvě metody. Obě dvě slouží pro dotazování se z databáze na data pro id školy a název školy. U metody „select\_obor“ je to stejné. Vše se vrátí metodou getResult().

Obrázek 22 metody select\_skola a select\_obor

```
public function select_skola(){
    $select = $this->db->query('SELECT skola.id, skola.nazev FROM skola ORDER BY `nazev` ASC;');
    return $select->getResult();
}

public function select_obor(){
    $select = $this->db->query('SELECT DISTINCT pocet_prijatych.obor, obor.nazev FROM pocet_prijatych
    INNER JOIN obor ON pocet_prijatych.obor=obor.id
    ORDER BY obor.nazev ASC;');
    return $select->getResult();
}
```

### 3.2.14 Model Skola

V modelu škola jsem vypsál jen město a jeho název. Jelikož si člověk chce vybírat podle názvu města a ne si pamatovat číslo, tak jsem spojil dvě tabulky a přidal k dotazu DISTINCT, aby se nezobrazovaly žádné výsledky dvakrát.

```
public function select_mesto(){
    // $db = db_connect();
    // $builder = $this->db->table('skola');
    // $query = $this->builder()->distinct()->select('skola.mesto, mesto.nazev')->join('mesto', 'skola.mesto=mesto.id')->orderBy('mesto.nazev','ASC');
    $select = $this->db->query('SELECT DISTINCT skola.mesto, mesto.nazev FROM skola INNER JOIN mesto ON skola.mesto=mesto.id ORDER BY mesto.nazev ASC;');
    return $select->getResult();
    //return $query->get()->getResult();
}
```



Obrázek 23 Ukázka v php

```

<label>Vyberte obec:</label>
<select name="mesto" class="form-control">
    <?php
    foreach($select_mesto as $row)
    {
        echo '<option value="'. $row->mesto. '">'. $row->nazev. '</option>';
    }
    ?>
</select>

```

### 3.3 Routes

Routes slouží k rychlejšímu načítání stránky, jelikož server potom nemusí skenovat další stránky, ale taky je to pracnější. Zde můžeme vidět metody ve formátu „\$routes->použitá metoda(‘stránka‘, ,‘JménoControlleru::funkce‘);“.

Používám metodu GET, POST, nebo metodu podobnou POST s názvem PUT, která je speciální pro codeIgniter. V routes jsou i tzv groups (skupiny). Zde jsem linkoval všechny metody pro přihlašování, odhlašování a další pro IonAuth, jelikož na internetu není žádný podrobný návod, jak implementovat IonAuth systém. IonAuth jsem připojil externě, takže je umístěn jako složka v kořenové složce webu. Všimněte si, že je IonAuth je zařazen v tzv. skupině (group).

Obrázek 24 Routes

```

$routes->get('/', 'Pages::index');
$routes->get('school_add', 'Pages::create');
$routes->post('school_add', 'Pages::school_add');
$routes->get('admitted_add', 'Pages::pocet_prijatych');
$routes->post('admitted_add', 'Pages::admitted_add');
$routes->get('maps', 'Pages::map');

$routes->get('school_edit/(:num)', 'Pages::school_edit/$1');
$routes->put('school_edit/(:num)', 'Pages::school_update/$1');
$routes->get('admitted_delete/(:num)', 'Pages::admitted_delete/$1');

$routes->group('auth', ['namespace' => 'IonAuth\Controllors'], function ($routes) {
$routes->get('/', 'Auth::index');
$routes->add('login', 'Auth::login');
$routes->get('logout', 'Auth::logout');
$routes->get('forgot_password', 'Auth::forgot_password');

$routes->get('create_group', 'Auth::create_group');
$routes->post('create_group', 'Auth::create_group');

$routes->get('create_user', 'Auth::create_user');
$routes->post('create_user', 'Auth::create_user');

$routes->get('edit_user/(:num)', 'Auth::edit_user/$1');
$routes->post('edit_user/(:num)', 'Auth::edit_user/$1');

$routes->get('edit_group/(:num)', 'Auth::edit_group/$1');
$routes->post('edit_group/(:num)', 'Auth::edit_group/$1');

$routes->get('activate/(:num)', 'Auth::activate/$1');
$routes->post('activate/(:num)', 'Auth::activate/$1');

$routes->get('deactivate/(:num)', 'Auth::deactivate/$1');
$routes->post('deactivate/(:num)', 'Auth::deactivate/$1');
});

```

IonAuth, jelikož se načítá externě z kořenové složky, musíme ho zapsat do „Autoload.php“ souboru. První je jméno čili „IonAuth“ poté následuje jméno složky „ion-auth“

```

public $psr4 = [
    APP_NAMESPACE => APPPATH, // For custom app namespace
    'Config'      => APPPATH . 'Config',
    'IonAuth'     => ROOTPATH . 'ion-auth',
];

```

## 4 UŽIVATELSKÁ DOKUMENTACE

### 4.1 Instalace na lokální Apache serveru

Web lze jednoduše rozjet na vlastním serveru za pomoci programu XAMPP. Ke stažení zde:

<https://www.apachefriends.org/download.html>.

První ale musíme stáhnout celou složku z githubu a extrahovat ji do složky htdocs programu XAMPP. Poté v phpmyadminovi (ve výchozím nastavení localhost/phpmyadmin) vytvořit novou databázi s kódováním „utf8\_general\_ci“ a poté vložit jako databázový dotaz celý soubor „mapy.sql“ ve složce „škola-mapy/database.“ Posledním krokem je si stáhnout composer odkaz na stažení: <https://getcomposer.org/download/>. Otevřít složku a v průzkumníkovi v horním políčku, kde je cesta k souboru napsat „cmd.exe.“ Napsat composer update a chvíli počkat, až se dostahování balíčků dokončí. Poté stačí složku pojmenovat „skola-mapy“ a je hotovo. Už jen spustit přes XAMPP (MySQL a Apache server).

### 4.2 Popis webu

Na hlavní stránce se nachází tabulka a tlačítka pro editování škol a jejich informací, Delete čímž tlačítko smažeme a vyhledávací pole „search“. Ve vyhledávacím poli se dají hledat jednotlivé údaje, v tabulka dokáže filtrovat podle jednoho kritéria, které si vyberete, stačí kliknout.

Na hlavní stránce se zobrazí školy jen pokud byly přidány dodatečné údaje o přijatých žácích, takže školy se můžou zdát duplikátní, ale ve skutečnosti zobrazuje školy, jejich obory a přijaté žáky.

Pro přidávání informací se musíme přihlásit. Kliknutím na tlačítko Login. Základní údaje pro přihlášení jako administrátor (pod kterým musíte být přihlášení, jinak byste nemohli nic přidávat) jsou: email: „admin@admin.com“ a heslo je: „password“ (vše bez uvozovek). Pokud se uživatel chce odhlásit, klikne na tlačítko logout.

Obrázek 25 hlavní stránka

Škola mapy   Databáze škol   mapy   Přijetí studenti   Login   Logout

Přidej školu Přidej školu

Show 5 entries Search:

ID Škola	Obor	Počet přijatých žáků	Město	Škola	Rok přijetí	Lattitude	Longitude	Akce
1	IT	1	Uherský Brod	ZŠ Na Výsluní Uherský Brod	2019	49.032687	17.643536	<span>Edit</span> <span>Delete</span>
2	OA	1	Uherský Brod	Katolická základní škola Uherský Brod	2019	49.022996	17.649707	<span>Edit</span> <span>Delete</span>
3	IT	4	Uherský Brod	ZŠ Mariánské náměstí Uherský Brod	2019	49.026235	17.645464	<span>Edit</span> <span>Delete</span>
3	OA	4	Uherský Brod	ZŠ Mariánské náměstí Uherský Brod	2019	49.026235	17.645464	<span>Edit</span> <span>Delete</span>
4	IT	3	Uherský Brod	ZŠ Pod Vinohrady Uherský Brod	2019	49.026792	17.653584	<span>Edit</span> <span>Delete</span>

Showing 1 to 5 of 58 entries Previous 1 2 3 4 5 ... 12 Next

#### 4.2.1 Přidávání škol

Zde můžete napsat název školy, vybrat město ze seznamu dostupných měst. A informace o šířce a délce s desetinnou tečkou např. „49.2342.“ Poté stačí kliknout na tlačítko „Uložit.“

Obrázek 26 Přidání školy

Přidej školu Back

Název školy

Zadejte název školy

Vyberte obec:

Babice

Geo\_latitude

Zadejte zeměpisnou šířku

Geo\_longitude

Zadejte zeměpisnou délku

Uložit

Obrázek 27 Úspěšné přidání školy

Úspěšně přidáno ×

**Přidej školu**
Přidej školu

Show 5 entries
Search:

ID Škola	Obor	Počet přijatých žáků	Město	Škola	Rok přijetí	Lattitude	Longitude	Akce
1	IT	1	Uherský Brod	ZŠ Na Výsluní Uherský Brod	2019	49.032687	17.643536	<span style="background-color: #4caf50; color: white; padding: 2px 5px;">Edit</span> <span style="background-color: #f44336; color: white; padding: 2px 5px;">Delete</span>
2	OA	1	Uherský Brod	Katolická základní škola Uherský Brod	2019	49.022996	17.649707	<span style="background-color: #4caf50; color: white; padding: 2px 5px;">Edit</span> <span style="background-color: #f44336; color: white; padding: 2px 5px;">Delete</span>
3	IT	4	Uherský Brod	ZŠ Mariánské náměstí Uherský Brod	2019	49.026235	17.645464	<span style="background-color: #4caf50; color: white; padding: 2px 5px;">Edit</span> <span style="background-color: #f44336; color: white; padding: 2px 5px;">Delete</span>
3	OA	4	Uherský Brod	ZŠ Mariánské náměstí Uherský Brod	2019	49.026235	17.645464	<span style="background-color: #4caf50; color: white; padding: 2px 5px;">Edit</span> <span style="background-color: #f44336; color: white; padding: 2px 5px;">Delete</span>
4	IT	3	Uherský Brod	ZŠ Pod Vinohrady Uherský Brod	2019	49.026792	17.653584	<span style="background-color: #4caf50; color: white; padding: 2px 5px;">Edit</span> <span style="background-color: #f44336; color: white; padding: 2px 5px;">Delete</span>

Showing 1 to 5 of 58 entries

Previous
1
2
3
4
5
...
12
Next

#### 4.2.2 Stránka přijatí studenti

K tomu, aby se zobrazila škola na hlavní stránce potřebujeme určit počet přijatých za školní rok. Zde vyplňujeme počet přijatých, školní rok a vybereme ze seznamu školu a obor.

Obrázek 28 počet přijatých na škole

**Počet přijatých na škole**
Back

Vyberte školu:

Academic School Uherské Hradiště

Počet přijatých:

Zadejte zeměpisnou šířku

Vyberte obor:

IT

Zadejte ročník

Zadejte zeměpisnou šířku

Uložit

#### 4.2.3 Úprava a mazání údajů o přijatých žácích

Mazání a úprava probíhá na hlavní stránce v tabulce, kde je zelené tlačítko „Edit“ pro úpravu dat a „Delete“ pro smazání dat přijatých žáků na školách, školy nemaže.

V editaci můžeme změnit název školy, souřadnice (Geo\_latitude, Geo\_longitude), obec. Pokud necháme obec prázdnou, nastaví se taková hodnota, jaká je tam nastavená.

Obrázek 29 editace školy

**Edituj školu**

Back

Název školy

ZŠ Mariánské náměstí Uherský Brod

Vyberte obec:

Select city

Geo\_latitude

49.026235

Geo\_longitude

17.645464

Uložit

## ZÁVĚR

Web zpracovaný v rámci předmětu webové aplikace. Využitý bude k maturitním projektu, jako ukázka nahrání dynamického webu.

Bohužel už se mi nepovedla spravit responzivita na první stránce, kdy navigace ustupuje směrem doleva.

## SEZNAM POUŽITÝCH ZDROJŮ

1. Youtube: #08 Removing public and index.php from URL | CodeIgniter 4 Tutorials [online]. [cit. 2021-11-25]. Dostupné z: [https://www.youtube.com/watch?v=Daw\\_bkWyP8o](https://www.youtube.com/watch?v=Daw_bkWyP8o)
2. Youtube: How To Edit / Update Data in Database Using Codeigniter [online]. [cit. 2021-11-25]. Dostupné z: [https://www.youtube.com/watch?v=Daw\\_bkWyP8o](https://www.youtube.com/watch?v=Daw_bkWyP8o)
3. Youtube: Part 10: Add and Edit User | CodeIgniter 3 and Bootstrap 3 [online]. [cit. 2021-11-25]. Dostupné z: [https://www.youtube.com/watch?v=\\_VIVtoHp9uk](https://www.youtube.com/watch?v=_VIVtoHp9uk)
4. Youtube: How to Edit Data in Database using Codeigniter 4 tutorial in Hindi [online]. [cit. 2021-11-25]. Dostupné z: <https://www.youtube.com/watch?v=wVNYDLY6N3A&t>
5. Youtube: CodeIgniter4 - Part-7: How to Edit & Update data by id using PUT method in codeigniter 4 [online]. [cit. 2021-11-25]. Dostupné z: <https://www.youtube.com/watch?v=59O3ZqqGwxA&t>
6. CodeIgniter Forum: [split] Ion-Auth with CI4 [online]. [cit. 2021-11-25]. Dostupné z: <https://forum.codeigniter.com/thread-74712-post-369353.html#pid369353>
7. GetBootstrap: Introduction [online]. [cit. 2021-11-25]. Dostupné z: <https://getbootstrap.com/docs/4.0/getting-started/introduction/>
8. GitHub: IonAuth v4 [online]. [cit. 2021-11-25]. Dostupné z: <https://github.com/benedmunds/CodeIgniter-Ion-Auth>
9. Leaflet: Download Leaflet [online]. [cit. 2021-11-25]. Dostupné z: <https://leafletjs.com/download.html>
10. CodeIgniter: CodeIgniter 4.1.5 docs [online]. [cit. 2021-11-25]. Dostupné z: <https://codeigniter4.github.io/userguide/index.html>
11. DataTables: DataTables download [online]. [cit. 2021-11-25]. Dostupné z: <https://datatables.net/download/>



## SEZNAM OBRÁZKŮ

Obrázek 1 controller a konstruktor .....	33
Obrázek 2 funkce index .....	34
Obrázek 3 funkce create.....	34
Obrázek 4 vyskakovací okénko "Úspěšně přidáno" .....	35
Obrázek 5 funkce pocet_prijatych .....	36
Obrázek 6 funkce admitted_add .....	36
Obrázek 7 funkce map .....	37
Obrázek 8 funkce school_update .....	38
Obrázek 9 funkce delete.....	38
Obrázek 10 model Main_model.....	39
Obrázek 11 metoda select_menu modelu Obor .....	40
Obrázek 12 metody select_skola a select_obor .....	40
Obrázek 13 Ukázka v php .....	41
Obrázek 14 Routes .....	42
Obrázek 15 hlavní stránka.....	44
Obrázek 16 Přidání školy .....	44
Obrázek 17 Úspěšné přidání školy.....	45